Converted by Andreas Bertelmann for ABBUC

Content

Chapter I

Preface

1 Preface

Why do we keep developing new products for the 8-bit Atari computers you ask? The easy answers are that we like what we are doing, we still have many positive ideas to contribute to the 8-bit Atari world, and developing these products from concept through production is really exciting! While those are partially the reasons, the true answer is that more Atari users are buying our products than ever before!

The ICD Multi I/O board (MIO) is a cumulation of all the great ideas you have asked us to develop. (Yes we do look at your suggestions.) It plugs into the parallel bus of the 800XL or the 130XE (with adapter) and provides: a RAMDISK, a printer port, a MODEM port, a printer buffer, a flexible hard disk interface, and built in configurations software to control it all. An optional 'hardware' 80 column adapter is also available. About the only other thing you could want is a floppy interface. With the hard drive and large RAMDISK it would be an unnecessary added expense. Most users already have one SIO floppy drive and even that will get little use after using the MIO.

When you fill out your warranty/update card you will also be added to our newsletter mailing list. This will keep you up to date on new products and developments as well as providing general information about our company and our full product line. MODEM users may also want to try out our informative ICD BBS. It is online 24 hours a day running 300/1200/2400 baud at (815) 968-2229. There are no charges for this service other than the standard long distance phone rates.

Chapter II

Introduction to the MIO

2 Introduction to the MIO

General Functions

The ICD Multi I/O board (MIO) is a complex peripheral device which adds exceptional power to the Atari 800XL or 130XE computer. The MIO gives the user five additional hardware functions not available with a stock 8-bit Atari computer. These are: a parallel printer interface, a serial interface for printer or MODEM, a printer buffer, a bootable RAMDISK, and an SCSI/SASI interface for hard disk drives. An optional sixth function (80 column board) may be added at a later date.

Parallel Port

This is a "centronics" type parallel port for connection to printers. The hardware connection uses a 15 pin sub D connector like the P:R: Connection or Atari 850 interface. All the pin connections are the same so the inexpensive P:R: Connection/850 cables will work to connect to your printer. See Chapter 2 and Appendix B for the complete pinout and connection diagram.

Serial Port

The serial port is an RS-232 type port for connection to serial printers, MODEMs or other computers. It uses a 9 pin sub D connector with the same pinouts as port 1 of the P:R: Connection or Atari 850 interface. See Chapter 2 and Appendix B for the complete pinout and connection diagram. The "R:" handler (in ROM) is equivalent to the P:R: Connection in compatibility, however, no memory inside your computer is used for the handler (since the "R:" handler is built into the ROM on the MIO). See Appendix C for complete compatibility information. You may also use the serial port to control a serial printer. Full XON/XOFF software handshake protocol and READY/BUSY hardware handshake protocol is supported through the "P:" handler on this port.

RAM

The RAM is available for use as RAMDISKs and/or a printer spooler (commonly called a printer buffer). Since the MIO has its own power supply and refresh oscillator, memory is retained even after powering down the computer. This unique feature along with drive swapping allows rebooting the computer directly from RAMDISK!

The MIO is available in 256K or 1 Meg versions. The RAM chips are not socketed and are a very special variety. The 256K versions are not user upgradeable! They may be upgraded by sending the MIO along with the correct upgrade fee to ICD's service department. (Call for current prices.) Any user modification attempts will void all warranties and result in additional service fees.

SCSI/SASI Interface

The SCSI/SASI interface allows the use of standard SCSI and SASI devices. These include hard disk drives, tape backup units, and whatever else is available with these protocols. Chapter 2 lists many of the current devices supported by this interface and the current software.

ROM

The ROM (16K bytes) in the MIO contains all the control software for the various features of the MIO. The built in MENU program is accessed by holding down the SELECT key while booting or by holding down SELECT and pressing RESET. Since the MENU loads into the computer's main memory (RAM), this action destroys the previous contents of program memory (computer RAM not the MIO RAM). To return to DOS (or cartridge) press RESET or '0'.

The MENU allows setting the configuration for your particular application and saving new default parameters to hard drive #0,0. This includes: assigning drives, partitioning them, setting the printer buffer, assigning ports, and selecting the R: handler if desired. One of the benefits of the MIO and the parallel bus is that the various handlers are MIO resident and do not need to load into machine memory as required by serial bus devices. This saves valuable memory since MEMLO remains unchanged.

Chapter III

General Installation and Power up

3 General Installation and Power up

This chapter contains all the necessary information to connect up the MIO to the various devices it supports. Once you have connected up these devices, you should read Chapter 3 on software configuration and Chapter 4 on hard disk configuration procedures.

130XE Adapter Board

130XE owners will need to use our special adapter board to use the MIO and replace the cartridge slot. This high quality board plugs directly into the cartridge and ECI connectors and converts all signals to be compatible with the 800XL bus while adding two additional cartridge slots.

These slots are connected in parallel so do not attempt to plug in two standard cartridges at the same time. The R-Time 8 may be plugged into one of the slots, and a standard cartridge may be plugged into the other slot. (You may also stack the two.) If using the SpartaDOS X cartridge with another cartridge (like a language cartridge), you must stack the other cartridge on top of the SpartaDOS X cartridge since it always controls the top cartridge. Note that the R-Time 8 is not really considered a cartridge, in the programming sense, since it does not use any cartridge area memory.

If you are using Atari brand cartridges with the 130XE Adapter, you will need to open the plastic shield which covers the cartridge edge connector. (Other brand cartridges do not have this shield.) A paper clip or pen point, pushed into one of the two release slots, will allow you to release this cover.

Parallel Printer Connections

Either parallel or serial printers may be used with the MIO. Most parallel printers will use the same cable as sold for the P:R: Connection (or Atari 850) parallel port. See Appendix B for pin connections used in these cables. The signals listed in table 2-1 are supported by the MIO. The parallel printer connector is on the right front side of the MIO and is labeled "PRINTER".

NOTE: The printer spooler will only support printers using one of these two ports. It will not spool text for an Atari brand direct connect printer.

Table 2-1. Standard Parallel Printer Signals

Direction	Function	Pin	Logic Level
from MIO	Data Strobe	1	Low true
to MIO	Busy	13	High true
to MIO	Fault	12	Low true
(none)	Ground	11	
from MIO	Data Bit 0	2	High true
from MIO	Data Bit 1	3	High true
from MIO	Data Bit 2	4	High true
from MIO	Data Bit 3	5	High true
from MIO	Data Bit 4	6	High true
from MIO	Data Bit 5	7	High true
from MIO	Data Bit 6	8	High true
from MIO	Data Bit 7	15	High true

MODEM Connections

Most MODEMs may be connected to the 9 pin serial port with our standard P:R: Connection (or Atari 850) MODEM cable. See Appendix B for pin connections used in these cables. Full hardware handshaking is supported from this connector and with the built in "R:" handler. The signals listed in table 2-2 are supported by the MIO. The serial connector is the 9 pin connector on the left middle of the MIO and is labeled "RS-232".

Table 2-2. The Most Common RS-232 Signals

Direction	Description	Abbv	Pin	Logic Level
from MIO	Transmitted data	XMT	3	high true
to MIO	Received data	RCV	4	high true
from MIO	Data terminal ready	DTR	1	high true
to MIO	Carrier detect	CRX	2	high true
to MIO	Data set ready	DSR	6	high true
from MIO	Request to send	RTS	7	high true
to MIO	Clear to send	CTS	8	high true
(none)	Signal ground	GND	5	

Note: All signals are high true (standard RS-232 definitions).

Serial Printer Connections

Serial printers will require the fabrication of a custom printer cable. All that is usually required for connection is: transmit, receive, and ground. Serial handshaking is done through software with XON/XOFF codes or with READY/BUSY hardware handshake. (READY/BUSY requires additional connections.) Refer to table 2-2 for the serial port pinouts on the MIO, and refer to your printer manual for the pinouts on its serial connector. Also, make sure that your printer supports XON/XOFF protocol. If it only supports a hardware handshake (READY/BUSY), then you can not use the printer buffer/spooler (see Chapter 3 about disabling the spooler). You must also connect CTS, DCD, and DSR, of the MIO to the "ready" pins of the printer (typically DTR). (The MIO checks CTS, DCD, and DSR, to make sure that they are high before sending each character.)

SCSI/SASI Connections

A 50 pin ribbon cable connects your SCSI or SASI controller board to the IO. Pin 1 is towards the rear of the MIO. Table 2-3 lists the signals used by this port. The SCSI/SASI connector is on the right rear of the MIO and is labeled "HARD DISK".

Table 2-3. Standard SCSI/SASI Signals

Direction	Description	Abbv	Pin	Notes
both	Data bit 0	-DB(0)	2	
both	Data bit 1	-DB(1)	4	
both	Data bit 2	-DB(2)	6	
both	Data bit 3	-DB(3)	8	
both	Data bit 4	-DB(4)	10	
both	Data bit 5	-DB(5)	12	
both	Data bit 6	-DB(6)	14	
both	Data bit 7	-DB(7)	16	
both	Parity bit	-PAR	18	N/C — not supported
from MIO	Attention	-ATN	32	N/C — not supported
to MIO	Bus Busy	-BSY	36	
from MIO	Acknowledge	-ACK	38	
from MIO	Bus Reset	-RST	40	
to MIO	Message phase	-MSG	42	Connected, not supported
from MIO	Select Target	-SEL	44	
to MIO	Command or Data	-C/D	46	
to MIO	Request Data	-REQ	48	
to MIO	Input or Output	-I/O	50	

Note: All signals are low true. All odd pins are connected to ground. -PAR and -ATN are not connected in the MIO but are not needed for bus operation.

The MIO and the hard disk controller are connected by a 50 conductor ribbon cable. (Maximum cable length is 20 feet.) If using multiple controllers, you must connect each controller onto the cable in a parallel fashion (simply snap another connecter on the cable and plug it into the controller).

You should use a termination resistor pack on the devices connected at both ends of the cable but not in between the ends. (The socket at U50 in the MIO contains its resistor pack.) It is absolutely essential that at least one device has a termination resistor pack, otherwise the driver circuits will fail to operate.

Each controller on the cable must have a unique bus ID. This is typically accomplished by placing one or several jumper plugs in particular location on the controller card. Bus IDs will either be numbered 0 through 7 or 01,02,04,08,10,20,40,80. In either case, make sure each controller has a unique bus ID. Note: on XEBEC 1410 controllers, the ID is 0 — there are no jumper options to select other bus IDs. (The newer XEBEC 1410A does have jumpers.) Adaptec 4000A and 4070 controllers use jumpers A-F to select bus ID. Use no jumpers for an ID of 0; jumper A-B for an ID of 1.

Hard Drive Interface to Controller

Some hard drives have controllers built in and can be recognized by a 50 pin SASI/SCSI interface on the drive itself. Drives in this category include: Seagate ST-225N, Rodime 650/652, and Iomega Alpha/Beta drives. Note that you may set the bus ID on these drives, but the LUN (logical unit number) is always 0.

Some embedded SCSI drives (Supra uses these) may have trouble booting the configuration sector from the hard drive. Symptoms are: the drive will not come up to speed until both the MIO and the computer are powered on. By this time it is too late the computer tries to read the boot sector before it is up to speed. The controller just NAKs to the MIO or the MIO times out which makes it appear as if no drive is present. A solution to this is to bend down pin 40 of the SCSI connector of the MIO. This disables the RESET line to the drive which is the problem with startup on some imbedded SCSI drives. Looking directly at the SCSI connector on the MIO, pin 40 is in the bottom row of pins and in six pins from the left side.

Most commonly found hard drives do not have controllers built in. These drives must be ST-506/ST-412 compatible (all IBM drives are ST-506/ST-412 compatible — this is the most common drive interface). These drives are characterized by a 34 pin and a 20 pin edge connector. In order to operate this class of drives, you need a controller card (not an IBM compatible type) which has an SASI or SCSI interface. Typically, these controllers may be mounted right on top of the hard drive frame assembly with proper spacing and shielding.

The SASI/SCSI controllers, which include XEBEC 1410(A), Adaptec 4000A, and Adaptec 4070, usually contains two or more functional 20 pin connectors (SASI controllers may have three using one as a test port) which allow you to hook up multiple hard drives. The drive number jumpers on the drive itself, combined with the 20 pin connector you plug into, determines the Logical Unit Number (LUN). Adaptec 4000A and 4070 have a J0 connector for LUN 0 and J1 for LUN 1. Remember to set the drive jumpers (on the drive) to match the connector you plug into (consult your drive manual).

The controller's 34 pin edge card connector is connected in parallel to each drive to be used with that controller, while the 20 pin connecters are connected to only one drive each. (The 34 conductor ribbon cable must be terminated at each end of the cable so make sure resistor packs are on the extremities but not in the middle.) As mentioned earlier, each controller connected to the 50 pin cable must have a unique bus ID, however, you now have multiple LUNs. You must check the drives to make sure that each has a unique ID (0-3 typically) and that the drives are plugged into the correct 20 pin connector. Refer to your controller manual for this connection. Important: some drives as shipped contain a socket with break away tabs. Usually all 4 drive select lines are shorted together. This is fine for one drive systems, but when connecting two drives to one

controller, you must make sure that only one drive select is allowed to pass on each drive (thus 3 tabs need to be broken). Please refer to your drive manual before breaking any of these tabs. The following is a list of SASI/SCSI controllers we presently support:

Manufacturer	Model #	Interface	Drive type
Adaptec	ACB-4000A	SCSI	ST506/412
	ACB-4070	SCSI	ST506/412 (RLL certified)
Iomega	Alpha 10H	SCSI	(cartridge drive 10Mb)
	Beta xxx	SCSI	(cartridge drive xxMb)
Rodime	RO650	SCSI	(includes drive 10Mb)
	RO652	SCSI	(includes drive 20Mb)
Seagate	ST225N	SCSI	(includes drive 20Mb)
Western Digital	WD1002-SHD	SASI	ST506/412
XEBEC	1410	SASI	ST506/412
	1410A	SASI	ST506/412

Adaptec ACB4000A and ACB4070 controllers are available at competitive prices from ICD. Call for more information.

80 Column Video Interface

The MIO has a video port on its front left side. This is for use with an optional 80 column adapter which plugs on top of the MIO circuit board. The pin assignments are listed in the 80 Column Adapter Users Manual.

Power-up Sequence

The power-up sequence when using the MIO is as follows:

- Always turn on your hard disk unit first if applicable. It needs time to get up to speed before being accessed. (Usually around 30 seconds.)
- Turn the MIO on next. (Floppy drives, MODEMs and printers don't really matter as long as they are not being accessed.)
- Now turn on the computer.

The power off sequence is just the reverse; the computer is always turned off first. The power off sequence on the MIO and other devices doesn't really matter. Just remembe as a general rule: don't turn the MIO off without turning off the computer and don't turn the computer on without first turning on the MIO. **(In other words, the computer should never be on with the MIO connected and turned off.)**

The MIO draws between 1 and 2 watts of power while in its active state. This power consumption is minimal but does require some thought for ventilation. Do not place any equipment on top of the MIO which would prevent its heat dissipation. The MIO RAM will retain its memory as long as power is maintained.

If the system seems to lock up with the access LED on, first try pressing RESET. If that does not free the MIO, power down the system (first the computer, then the MIO), wait about 30 seconds for the memory to clear, and then power up the system again. Lockups can be caused by bad software or power surges.

Important: Under some conditions, when booting the computer, the MIO will take a long time to initialize. This is apparent when the MIO red light comes on but the screen is still black after several seconds of waiting. This is usually due to the MIO waiting for a non-existent drive to respond. The MIO attempts to communicate with each drive listed in the drive configuration, but if a drive is off-line, it may take some time for the controller to relay this information back to the MIO. Thus, it may take up to 30 seconds for the MIO to initialize.

Chapter IV

Software Configuration

4 Software Configuration

This chapter is solely devoted to the MIO software configuration program. The program MENU is entered by pressing (and holding) the SELECT key while booting the computer or while pressing RESET. When the menu appears, you may release the SELECT key.

Selecting Sub-Menus

The configuration menus are divided into four logical areas:

1. The upper left of the screen contains the sub-menu choices:

   1 Drive Configuration
   2 Printer Configuration
   3 RS-232 Configuration
   4 Save Configuration

   By pressing the keys '1' through '3', you may enter the desired sub-menu. Note that initially you enter the "Drive Configuration" sub-menu (characterized by a black screen). The sub-menu choices in the upper left remain the same through all three sub-menus. The fourth option, "Save Configuration" is not a sub-menu, you are simply asked if you want to save the configuration to your hard disk. (More on that later.)

2. The upper right of the screen contains the sub-menu options. These options affect the current sub-menu and allow you to change parameters in the middle of the screen (and bottom third if in the "Drive Configuration" sub-menu).

3. The middle third of the screen displays the current configuration or status which the current sub-menu allows you to change. If you perform a "Save Configuration", this data (from all sub-menu) is saved to the hard disk (ID=0, LUN=0). As you switch between sub-menus, this information will change to reflect the status of the new sub-menu.

4. The bottom third of the screen always shows the condensed drive configuration. This will display the type of drive (if any) that is assigned to each drive number (D1: through D8:).

```
      ICD Multi I/O Board

1 - Drive Conig      N - Drive Number
2 - Printer config   T - Drive Type
3 - RS-232 Config    S - Swap Drives
4 - Save Config      L - Write Lock

Selection:

         Drive # 1 Status

Drive Type: Hard     Cylinders:     615
Interface:  SCSI     Heads:         4
ID, LUN:    0,0      Start Sector:  1
Drive Size: 16562K   End+1 Sector:  65536

        Drive Configuration

D1: Hard (16562K)    D5:  Floppy #1
D2: Hard (4000K)     D6:  n/a
D3: RAM (750K)       D7:  n/a
D4: n/a              D8:  n/a
```

Figure 3-1. Sample MIO Configuration Display

Drive Configuration

This section describes the procedure for configuring your drives. Press '1' to enter the "Drive Configuration" sub-menu. The middle third of the screen shows the status of the selected drive number (1 through 8). The bottom third is a condensed version which shows drive type and size if applicable (floppy switch setting if a floppy drive).

N – Drive Number
Press 'N' to select the drive number. This allows you to cycle through the possible drive numbers and edit whichever drive you desire (with T, S, or L).

T – Drive Type
Press 'T' to change drive type. Now press SPACE to toggle through the possible choices: RAM Disk, Hard Disk, Floppy, or not assigned. Press 'RETURN' when the desired choice is displayed.

If selecting a RAM Disk, then press the space bar until the desired size appears (in increments of 32K), finally press RETURN when done. Note that the every RAM disk (in the MIO) is reformatted whenever you change a RAM Disk size or Print Spooler Size.

If selecting a floppy drive, you must then select the physical drive number of the drive (the number the drive would normally be accessed by). Note that if you choose "not assigned" ("n/a") for a particular drive number, the floppy drive connected as that drive number will respond. (e.g. Selecting "n/a" for D1: with a floppy #1 connected as any other drive number, will still allow access to floppy #1 but now as "D1:".)

If selecting a hard drive, you must follow the prompts with the following responses (follow each response with a RETURN): a) "Interface" — press SPACE until correct choice is displayed, b) "ID, LUN" — enter the correct SASI/SCSI bus ID (0-7), followed by the drive until number (0-7) (no RETURN necessary if both numbers entered, otherwise press RETURN), c) "Cylinders" — enter the number of cylinders the drive contains (or RETURN for no change), d) "Heads" — enter the number of heads your drive has (or RETURN for no change), e) "Start Sector" — enter first sector number in this partition (more on this later), f) "End+1 Sector" — enter the last (plus one) sector number in this partition (more on this later).

S – Swap Drives
Press 'S' to swap two drives. You will be asked "Exchange With D?" — enter the drive number you wish to swap the current drive with. This allows you to quickly move drives around. The primary use for this is to select desired hard drive to boot from (as drive 1) or to select a RAM disk to boot when system powered up.

L – Write Lock
Press 'L' to write lock current drive. This is the same as placing a write protect tab on a floppy. All writes to the drive are disabled. Note that only RAM drives and Hard drives will allow you to write protect them (cover the notch on floppies to protect them). This does not protect a hard disk from HDFMTPH.COM — it bypasses much of the configuration tables.

If setting up a hard drive, read Chapter 4 thoroughly before attempting the configuration menu. Briefly, there are 3 main steps to set up the hard drive: 1) set interface type, ID/LUN, cylinders, and heads, 2) format hard drives, 3) set partitions, and 4) build directory structures. Chapter 4 will take you step by step through this procedure. It will also discuss the meaning of the "Start Sector" and "End+1 Sector" numbers.

Printer Configuration

This section describes the procedure for configuring your printer. Press '2' to enter the "Printer Configuration" sub-menu. The middle third of the screen shows the current status of the printer spooler and options. The options sub-menu choices (in the upper right corner) are as follows:

P – Pause/Resume
Press 'P' to toggle paused state. When the spooler is paused, the printer will stop printing until you press another 'P' to continue printing. This is only applicable if using the spooler.

C – Clear Spooler
Press 'C' to clear the spooler. All pointers will be reset and the buffer emptied. This is only applicable if using the spooler.

R – Repeat Copies
Press 'R' to instruct the spooler to print another copy of the document just printed. This will reprint everything currently in the buffer. Make sure that before you print a document (that you want several copies of), that you clear the buffer first. You may stack up to 9 repeat copies. Again, this is only applicable if using the spooler.

T – Set Port Type
Continue pressing 'T' until the desired port type is displayed. This toggles through 2 parameters: the line feed option, and the port type (serial/parallel). If "Yes" to "Line Feeds", then a line feed will be printed after every carriage RETURN (EOL character on Atari). This is how most computers operate. If "No", then the printer must be set in a forced line feed following carriage return mode (usually by some internal switches on the printer). The other parameter "Port Type" determines which port will be used for the printer. If "Parallel", then the parallel port on the right of the MIO is used (this is the usual configuration). If "Serial", then the serial port on the left of the MIO is used. The RS-232 port supports XON/XOFF software handshake protocol and READY/BUSY hardware handshaking protocol for serial printers.

S – Spooler Size
Continue pressing 'S' until the desired printer spooler size is shown. The size is selected in 32k increments. If a size of 0 (Spooler: Off) is selected, then the buffer is disabled and the port acts like a normal P:R: Connection or 850 printer port. Note that the every RAM disk (in the MIO) is reformatted whenever you change a RAM Disk size or Print Spooler Size..

N – Port Number
Press 'N' until the desired port number is displayed. You have the option of letting the MIO act as printer 1, (1 is normal) 2, or none at all (printer port disabled). This will override all other external printer interfaces, however, if you select the MIO as "P1:", then printing to "P2:" will go to the external printer interface and vice versa.

RS-232 Configuration

This section describes the procedures for configuring your RS-232 port. Press '3' to enter the "RS-232 Configuration" sub-menu. The middle third of the screen (the status area) shows the current setting of the RS-232 port. The port configuration (baud rate, stop bits, and parity) is primarily used when configuring for a serial printer. When using the port through the "R:" handler, it is configured through XIO calls to the "R:" handler (by terminal programs and bulletin boards — BBS).

B – Baud Rate
Continue pressing 'B' until the desired baud rate is displayed. This value is a default only and can be overridden by an XIO to the "R:" handler (if enabled).

S – Stop Bits
Continue pressing 'S' until the desired number of stop bits is displayed. This value is a default only and can be overridden by an XIO to the "R:" handler (if enabled).

P – Parity Mode
Continue pressing 'P' until the desired parity mode is displayed. This value is a default only and can be overridden by an XIO to the "R:" handler (if enabled).

A – Port Assign
Press 'A' to change the port assignment. If the port is already assigned to the printer, then pressing 'A' does nothing. If the port is not assigned, then it may be assigned to the "R:" handler and used to interface to MODEMs or other computers.

Save Configuration

Press '4' followed by a 'Y' ("Are you sure?") to save the current configuration to the hard disk (Unit 0,0). Then, when you reboot the system (and the MIO has been powered off and back on again), your configuration will be read from the hard disk. This is so that you do not have to set the configuration each time you boot your system, however, as long as the MIO remains on, it will retain its current configuration. Thus, the configuration is read from the hard drive only when the MIO has lost power and you are rebooting your system.

Using SpartaDOS

SpartaDOS 3.x or higher supports parallel devices. You cannot use the MIO with any SpartaDOS version lower than that. SpartaDOS 3.2 supports up to 16 megabytes per drive and gives a MEMLO of $1A88 with the MIO installed. SpartaDOS is strongly recommended as the DOS to use with the MIO! We will not be responsible for problems arising from the use of other DOS types with the MIO.

Using Other DOS Types

Most DOSes (except MyDOS) will not support drives larger than 1024 sectors (about 256K). This is a limitation of the Atari DOS 2 compatible sector chain, not the MIO. MyDOS doesn't have this limitation, but it is still a small computer style of DOS. Simply put, you get more flexibility with SpartaDOS than any other DOS for the Atari 8-bit line of computers. (SpartaDOS is much like MS-DOS and UNIX — there must be a reason they are so popular.)

Since you have a lot of flexibility in drive assignments, you may still want to put DOS 2 (or MyDOS) on one of the partitions of a hard drive — this way you can swap drives and boot whichever DOS you would like.

Using Boot Disks

To use a boot disk, go to the configuration editor and set "D1:" as "Floppy #1". The best way to do this is to "Swap Drives" between "D1:" and the drive that says "Floppy #1" after it. If none are configured as floppy 1, then swap with a "n/a" drive. The computer can now be rebooted from floppy drive 1.

Chapter V

Setting Up a Hard Disk

5 Setting Up a Hard Disk

Hardware Considerations

The MIO includes a powerful built in SASI/SCSI interface. This allows you to connect any SASI (sassy) or SCSI (scuzzy) device to your Atari providing you have the proper support software. SASI (Shugart Associates Systems Interface), developed by Shugart first, is a subset of SCSI (Small Computer Systems Interface). SCSI is the more recently adopted "standard" for small computer systems other than IBM. SASI/SCSI controllers typically interface with an ST-412 or ST-506 hard drive. ST-412 or ST-506 is the standard interface used on most 'bare bones' 5 1/4 inch hard drives in use today. The MIO will physically (electronically) work with all SASI/SCSI controllers but we cannot guarantee to have format software for every possible configuration. (SASI controllers usually use the standard SASI format routine. SCSI controllers typically need special routines for each brand.)

Controllers not supported with format routines

If you have a SASI or SCSI controller or drive not supported by our HDFMTPHx.COM program, we may modify our format routine to handle it. You must send us your controller (or drive) and its technical manual along with our special service fee of $45. Please call our technical support department before sending in your system. We may already have the necessary format routine written.

Initialization Sequence if Using SpartaDOS on your hard drive.

The following is a brief description of the steps necessary to set up a hard disk drive using SpartaDOS 3.2.

1. Connect the hardware - includes use of proper cables, proper termination, and drive ID assignments. For more information, refer to Chapter 2.

2. Set MIO configuration - includes interface types, heads, cylinders, and ID/LUN. This is how HDFMTPH.COM will find your hard drive. If you are going to partition your hard drive, you must set the configuration for each drive unit number ("D1:" - "D8:") and the parameters must match. Refer to Chapter 3 for configuration instructions.

3. Physically format the disk - use HDFMTPH.COM and make proper menu selections for your controller. Note that if partitioning a drive, several drive numbers will follow the drive number selection. Be careful since you do not enter the drive number but rather a selection number.

4. Partition the drive - only necessary if you made two or more drive numbers ("D1: - "D8:") point to the same physical drive. The partitions are set up by simply giving a start and end sector number (the HDFMTPH.COM program displays the total number of sectors on the hard drive) in the MIO Configuration Menu. Read the next section for more information on setting Start/End sector numbers.

5. Save the configuration by pressing '4' and then 'Y' in the MIO Configuration Menu. Note: the configuration is written to sector 0 of the drive with the ID/LUN of 0,0.

6. Directory build - use HDFMTDI.COM which writes the SpartaDOS directory structure to the hard disk. This should be repeated on each drive partition. When asked which drive number, enter 1 through 8 (for "D1" through "D8:"). This is not to be confused with either the bus ID of the drive or the drive selection number in the HDFMTPH.COM program.

7. Copy SpartaDOS 3.2 to each partition you plan on booting from. (Simply use the COPY command — this is documented in the SpartaDOS Construction Set Manual.) Next, use the BOOT command to assign the DOS as the boot program. (Type "BOOT Dn:X32D.DOS" where 'n' is the drive you copied DOS to — refer to the SDCS manual for more information on the BOOT command — this is about the least understood and easiest to use command in SpartaDOS.)

Initialization Sequence if Using MyDOS on your hard drive.

The following is a brief description of the steps necessary to set up a hard disk drive using MyDOS.

Follow steps 1-5 of the above sequence. Boot MyDOS and enter the menu. Set the drive configuration by pressing 'x' and answering 'Y' to the large capacity drive question and enter the total number of sectors (as displayed by HDFMTPH.COM). Next press 'I' (followed by the drive number) to initialize the directory, and finally use the 'H' command to write DOS files to that drive.

Formatting Your Hard Disk — More Complete Details

The following is a more detailed description of the processing of setting up your hard drives system. Up to this point, enough information has been given, but it may still not make sense. Therefore, the following goes through the process again attempting to make everything more clear.

System Configuration
The system configuration must be setup before attempting to format your hard disk drive. Hold down the SELECT key and press RESET. You can release the SELECT key after the configuration menu appears. Device ID, Logical Unit Number, Heads, and Cylinders, must be set before performing the physical format. These tell the system drive location and size. Chapter 3 documents the configuration menu in detail in case you are not sure what to do.

Device ID
This is the SASI/SCSI bus ID. Jumper plugs (shorting type) on the controller are used to set the device ID. Set this to drive #0 if only using one drive. (0 is the only drive for loading and saving default configurations.) Values of 0 through 7 are valid. Device ID and Logical Unit Number (ID,LUN) are used so the system can identify individual physical drives. D1:, D2:, etc. are not valid ways to identify a particular drive anymore since the MIO allows you to swap drive numbers and partition a physical drive into more than one drive.

Logical Unit Number
The logical unit number (LUN) is determined by which 20 pin plug on a SASI/SCSI controller is used and by the drive number the hard disk itself is configured to. This will be 0 or 1 on most systems. Refer to your controller manual for plug locations. (One drive with an ID of 0 must be plugged into LUN 0 since that is where the configuration is loaded and saved.)

Heads and Cylinders
Refer to your hard drive manual for number of heads and cylinders. You may also refer to Appendix D which is a general chart of some common hard drives. Heads refers to the actual number of read/write heads inside your drive and cylinders are similar to tracks on a floppy disk. A typically 10 Meg drive may contain 4 heads and 306 cylinders or 2 heads and 612 cylinders.

Format (HDFMTPH.COM)
The physical format is created and verified by running the HDFMTPH.COM program. If using SpartaDOS, run this as a command file. (With another DOS, run this as a binary file.) Then select the proper interface type from the menu. CAUTION: Running the physical format totally destroys all previous information on the drive! This process is only needed the first time you set up your drive. A 20 Meg physical format will take about 30 minutes so be patient. When finished, the total number of available sectors will be displayed. Make sure you write this number down for each drive. This is the only time you will be told how many good sectors are available for partitioning. Defective sectors will be mapped out automatically by the controller. If the drive is not to be partitioned, then HDFMTPH.COM will automatically set up the Start/End+1 sector numbers in the configuration menu.

Partitioning
The hard drives must now be partitioned. The MIO is simply looking for a starting and ending sector for each drive. SpartaDOS can handle up to 16 Meg (16,776,960 bytes or 65535 sectors) per drive. Other DOS types may have smaller limitations. Check with the DOS publisher for specification. The format uses 256 byte double density sectors.

Assuming a 16 Meg limit per partition, if the starting sector is 1 (0 is always reserved), the maximum ending sector +1 should be 65536. The next partition should start with 65536 and end with a maximum ending sector number of 131071. (Add 65536 to get the next maximum.)

Directory Build (HDFMTDIR.COM)
HDFMTDIR.COM (directory build) writes the SpartaDOS directory structure to the disk drive. Run this command if you are using SpartaDOS as your system DOS. This command will also destroy any previous data on the partition of the hard disk that is written to.

If using MYDOS or other DOS types you will use the INIT functions and Write DOS functions instead of HDFMTDIR. See the DOS manual for more information.

Directory build should be run on each drive (D1:, D2:, etc.) used on the hard drive. The SpartaDOS module should then be copied to the partition to be used as D1: and the BOOT command used to assign the boot flag.

Example: BOOT X32D.DOS <return> (Assuming version X32D.DOS.)

Chapter VI

MIO Hardware Configuration

6 MIO Hardware Configuration

This chapter fully documents the operation of the MIO hardware. This information is intended for the curious and those who wish to better utilize the power of the MIO. If you do plan on accessing the hardware directly, you must take care since the software (in ROM) has set several conventions that must be followed if any portion of its software is to be used. This information is in the section entitled "Software Description".

Hardware Description

The MIO has 3 basic addressing areas: 1) ACIA at $D1C0-$D1DF, 2) MIO Latches at $D1E0-$D1FF, and 3) 256 bytes of RAM at $D600-$D6FF. Note that the ACIA and MIO latches are not fully decoded; both contain 4 read/write registers, but each register has 7 shadows. Tables 5-1 through 5-3 briefly describes the MIO registers.

ACIA Operation

For more information on the ACIA operation, refer to a 6551A or 65C51 data manual (this part is manufactured by Rockwell, GTE, NCR, and RCA). The only irregularity of usage is that the lines DSR, CTS, and DCD are tied to ground. This is due to the fact that the ACIA will not receive data if either of these lines are false. To read the actual state of these lines, you must read location $D1E3 (bits 2,1,0). This will return the true lines sense (DCD true is indicated by a high on bit 1 of $D1E3).

SASI/SCSI Interface

The data input/output ($D1E1) and input control lines ($D1E2) reflect the true voltage levels on the ports. Thus, the input control lines are normally all ones (port voltages are +5) which represents a logic false on the bus. For more information on SASI/SCSI bus protocol, refer to the Adaptec ACB-4000 Series User's Manual, the XEBEC S1410A disk controller document, the Seagate ST225N manual, or any other device manual employing these protocols.

The ACK-/REQ- handshake cycle is performed by the MIO hardware. Whenever the data ($D1E1) is read or written, ACK- is set true. It is cleared by a high level on the REQ- input signal.

The RST- is set true when $D1E0 is read or when the RESET key pressed on the computer (or during powerup). It is cleared by reading location $D1E2.

The I/O- controls whether the 8-bit printer/SASI registers are input or output. When I/O- is high (input to SASI controller), the data is output on the printer and SASI port. If I/O- is low, then the output latches are disabled (it will latch new data, but the output is tri-stated).

Printer Interface

The printer data ($D1E1) and BUSY signals are high true logic. The a high (1) on BUSY (bit 6 of $D1E2) indicates the printer is busy. A low (0) on FAULT- (bit 4 of $D1E2) reflects an error condition in the printer (printer off?).

Table 5-1. MIO Register Selection

Address (HEX)	Register Operation	
	Write	Read
$D1E0	Set address A15-A8 for $D600 RAM window. (LSB of sector number.)	Set RST- signal true (low). Resets the SCSI/SASI bus. (RST- also true during RESET)
$D1E1	Set printer data and SCSI/SASI data. True logic for printer — Inverted for SCSI/SASI bus.	Read data from SCSI/SASI bus. Data is inverted.
$D1E2	General purpose outputs. B[3..0] High RAM address, sets address A19-A16. B[4] 1 = Set SEL- true B[5] 1 = Enable RAM access B[6] 1 = Set STROBE- true B[7] 1 = Enable Parallel IRQ	General purpose inputs. B[0] = SASI C/D- B[1] = SASI MSG- B[2] = SASI I/O- B[4] = Printer FAULT- B[5] = SASI BUSY- B[6] = Printer BUSY B[7] = SASI REQ- Also clears RST- signal
$D1E3 or $D1FF	Set ROM enable and bank. Only 1 bit allowed set at a time. B[2] 1 = Disk Interface ROM B[3] 1 = Seg 2 of setup MENU B[4] 1 = R:/P: Handler ROM B[5] 1 = Seg 1 of setup MENU All bits 0 disable the ROM.	IRQ sense bits + Misc inputs. B[0] = RS-232 DCD line B[1] = RS-232 DSR line B[2] = RS-232 CTS line B[3] = Printer BUSY- IRQ B[4] = MIO IRQ (from 6551 or Printer BUSY- IRQ)
$D1C0	Write ACIA transmit register.	Read ACIA receive register.
$D1C1	Perform a programmed RESET on ACIA (data is "don't care").	Read Status register (resets IRQ). B[0] 1 = Parity error B[1] 1 = Framing error B[2] 1 = Overrun has occurred B[3] 1 = Receiver reg. full B[4] 1 = Transmitter empty B[7] 1 = IRQ occurred
$D1C2	Write ACIA command register. (see table 5-2)	Read ACIA command register. (see table 5-2)
$D1C3	Write ACIA control register. (see table 5-3)	Read ACIA control register. (see table 5-3)
$D6xx	Write RAM. High address A19-A8 selected by $D1E0/$D1E2.	Read RAM. High address A19-A8 selected by $D1E0/D1E2.

Table 5-2. ACIA Command Register

```
 7     6     5     4     3     2     1     0
   PMC
PMC1  PMC0  PME   REM   TIC1  TIC0  IRD   DTR
```

Bits 7-6 Parity Mode Control (PMC)

```
7     6
O     0 Odd parity transmitted/received.
O     1 Even parity transmitted/received.
```

```
1     0 rAark parity bit transmitted, parity check disabled.
1     1 Space parity bit transmitted, parity check disabled.
```

Bit 5 Parity Mode Enable (PME)

```
O     Parity mode disabled -- no parity bit transmitted0
1     Parity mode enabled.
```

Bit 4 Receiver Echo Mode (REM)

```
0     Receiver normal mode.
1     Receiver echo mode -- bits 2 and 3 must be zero for receiver echo
mode, RTS will be true.
```

Bit 3-2 Transmitter Interrupt Control (TIC)

```
3     2
O     0 RTS    = false, transmitter disabled.
o     1 RTS    = true, transmit interrupt enabled.
I     0 RTS    = true, transmit interrupt disabled.
1     1 RTS    = true, transmit interrupt disabled,
                 and transmit break on TxD.
```

Bit 1 Receiver Interrupt Request Disabled (IRD)

```
0     IRQ- enabled (receiver)
1     IRQ- disabled (receiver)
```

Bit 0 Data Terminal Ready (DTR)

```
0     Data terminal not ready CDTR false)*.
1     Data terminal ready (DTR true).
```

Note
* The transmitter is disabled immediately. The receiver is disabled but will first complete receiving the byte in process.

Table 5-3. ACIA Control Register

7	6	5	4	3	**2**	1	0
	WL		SBR				
SBN	WL1	WL0	RCS	SBR3	SBR2	SBR1	SBR0

Bit 7 Stop Bit Number (SBN)

```
0     1 Stop bit.
1     2 Stop bits.
      1 1/2 stop bits for WL=5 and no parity.
      1 stop bit for WL=8 and parity.
```

Bits 6-5 Word Length (WL)

6-5	Number of bits
0 0	8
0 1	7
1 0	6
1 1	5

Bit 4 Receiver Clock Source (RCS)

```
0     External receiver clock (non-functional on MIO).
1     Baud Rate (SBR).
```

Bits 3-O Selected Baud Rate (SBR)

```
0     0     0     0 16 x RxC (Not usable by MIO).
o     0     0     1 50
0     0     1     0 75
o     0     1     1 110
0     1     0     0 135
```

```
0     1     0     1 150
0     1     1     0 300
0     1     1     1 600
1     0     0     0 1200
1     0     0     1 1800
1     0     1     0 2400
1     0     1     1 3600
1     1     0     0 4800
1     1     0     1 7200
1     1     1     0 9600
1     1     1     1 19200
```

Addressing the RAM

The MIO can access up to 1 Megabyte of RAM which takes 20 bits to address. Address bits A19-A16 are set from writing to the latch at $D1E2, bits A15-A8 are set from writing to the latch at $D1E0, and bits A7-A0 are CPU address lines A7-A0 when reading/writing $D6xx. Thus there are up to 4096 "pages" of memory that may appear at the $D6xx window.

In order to access the memory, it must first be enabled by setting $D1E2 bit 5 to "1" (this also turns on the MIO's red LED). It is generally a good idea to leave the RAM disabled while not using it in case of a system crash (which could inadvertently write in the $D6xx window).

When power is removed from the computer (for whatever reason), the MIO will continue refreshing its dynamic RAM. This is accomplished by its ability to maintain a 02 clock after the computers clock has stopped. VC1 adjusts the MIO's 02 clock frequency. Adjustment requires special equipment and should not be attempted.

Checking IRQ Status

The MIO has two sources of interrupts; one is the ACIA and the other is the parallel printer port. The printer port may interrupt the computer only if bit 7 of $D1E2 is set ('1') and the printer BUSY is false ('0'). Bit 4 of $D1E3 is the general IRQ flag from the MIO (a 1 indicates that IRQ- is true). If bit 3 is also set, then the IRQ- is caused by the printer. If not, then it must be the ACIA (in which case $D1C1 bit 7 should be set).

Note that the parallel device IRQ mask (PDIMSK at $249) is set to $10 by the MIO RAM. This is because, there is only one interrupt handler (which supports all possible MIO interrupts) in the ROM. In fact, the system would crash if the OS tried to enter any of the other ROM banks to service the IRQ.

Accessing the ROM

The ROM on the MIO contains all the software necessary to access the hard disk, the RAM, the ACIA (as an R: or P:), and the parallel printer port. It also contains the configuration which is downloaded into the computer RAM when SELECT+RESET are pressed.

The ROM is accessed as 4-2K banks. (An additional 8K is reserved for the 80 column adapter.) Bits 5-2 (of $D1E3) select which bank will be active (if any) at the $D800-$DFFF region. Only 1 bit may be set and its position selects which bank of ROM is active. If all bits are zero, then no banks are active and the Floating Point Math package in the OS ROM is enabled.

According to Atari spec, 1 device is to occupy one bank of ROM and that device has a specific address range legal to it at $D1xx. Since the MIO is an all inclusive device, however, it deviates from this spec. Instead, it tries to cram as much code as possible into a small space. This meant juggling the banks around to get along with the computer and to allow for expansion of an 80 column adapter. This is why there is only one interrupt handler, yet several input bits are returned in what is considered to be strictly an interrupt sense register (at $D1FF).

Software Description

In order for the MIO to perform its multitude of tasks, one full page of memory has been allocated for general operating variables and configuration parameters. Two other pages have been reserved for "R:" handler input and output buffers, and the rest of memory (up to 4093 pages) can be used as RAM drives and a printer buffer through the MIO ROM. Table 5-4 describes the configuration parameters (those which are read from the hard disk (ID=0, LUN=0) when memory is invalid). This table occupies the first 192 bytes of memory page 0. The remaining 64 bytes are operating variables and are listed in table 5-5. Memory pages 1 and 2 are reserved for the "R:" handler input/output buffers.

Table 5-4. MIO Configuration Parameters

Address (HEX)	Symbol Name	Length (Dec)	Function of parameter or variable.
$D600	MEMKEY	16	This contains a string of characters. If the string in memory is not equal to that in ROM, it is assumed that power to the MIO has been lost and it should reconfigure.
$D610	DRDATA	64	This contains an array of 8 drive config records (each 8 bytes long). Records are: +0: First physical block address (sector number) of logical device. MSB first. +3: Last+1 physical block address (sector number) of logical device. MSB first. +6: B[2,0] = SCSI/SASI ID if hard disk = drive number if floppy B[5] = 1 if floppy drive (reassign) B[6] = 1 if RAM drive B[7] = 1 if Hard drive if B[7,5] = 0, then ignore +7: B[3] = 1 if disk is write locked B[4] = 1 if SASI type interface B[7,5] = logical unit number of drive
$D652	PREND	2	Last+1 RAM page number allocated to print spooler.
$D654	PRUNIT	1	Printer device number (0 if P: disabled)
$D655	PRFLAGS	1	Printer configuration flags B[5] = 1 if using a serial printer B[6] = 1 if spooler is enabled B[7] = 1 if CR/LF option enabled
$D656	SERUNIT	1	RS-232 "R:" enable flag; 1=enable/0=disable
$D657	SERFLAGS	1	Default configuration for serial port. B[7] = 1 if to append LF after CR B[6] = 1 if 2 stop bits (else 1 stop bit) B[5] = 1 if no ATASCII/ASCII translation B[4,3] = parity mode: 00=none, 01=odd, 10=even, 11=mark B[2,0] = baud rate index
$D658	RAMUSAGL	8	Number pages allocated for each drive (low)
$D660	RAMUSAGH	8	Number pages allocated for each drive (hi)
$D668	RAMSIZE	1	Total number of RAM pages (high byte)
$D680	DRITYPE	64	Configuration data for SASI hard drives +0: Number cylinders on drive (MSB,LSB) +2: Numer heads on hard drive +3: Cylinder to start reduced write current +5: Precompensation value (usually 0) +7: ECC burst length (usually $0B)

Table 5-5. MIO Operating Variables

Address (HEX)	Symbol Name	Length (Dec)	Function of operating variable
$D6C1	CPRINPG	2	Printer queue character entry pointer. (memory page number) (LSB,MSB)
$D6C3	PRINPG	2	Printer queue character exit pointer. (memory page number) (LSB,MSB)
$D6C5	CPROFFS	1	Printer queue entry page offset.
$D6C6	PROFFS	1	Printer queue exit page offset.
$D6C7	BADBUFF	1	If 255, this indicates that the queue has wrapped, thus repeat copies are invalid.
$D6C8	PRIRQ	1	If 128, then parallel IRQ is enabled. This byte gets copied to $D6E2 when ROM exited.
$D6CA	PRPAUSE	1	If 0, then the printer spooler is paused.
$D6CB	PCOPYT	1	Number repeat copies to be printed (normally zero unless they get stacked)
$D6CC	SPOOLGO	1	Master spooler start/stop flag (255=go)
$D6CD	XON	1	XON/XOFF flag for serial printer handshake. 255=on 0=off.
$D6FC	CURPAGE	1	Shadow for $D6E0. Needed for IRQ operation.
$D6FD	SMISC	1	Shadow for $D6E2. Needed for IRQ operation.

Chapter VII

RS-232 Handler Functions and Tables

7 RS-232 Handler Functions and Tables

This chapter contains a list of all input/output and XIO calls to the RS-232 "R:" handler of the MIO. Note that IOCB is an input/output channel number that indicates what OPEN device shall receive or provide data. For most XIO calls, you may use any legal IOCB number as long as it is NOT open to any other device. From Atari BASIC, you may use IOCB numbers 1 through 7 (0 is reserved for editing "E:" I/O).

Note that IOCB #7 is used for the BASIC LPRINT statement and IOCB #6 is used for graphics mode functions from BASIC. Also, if using SpartaDOS, IOCB #4 and IOCB #5 are used while doing output and input redirection respectively (via the DOS PRINT command and batch files).

All the syntaxes use just "R:" for the device name since there is only one RS-232 port on the MIO. In fact, if you do use a port number (ex. "R2:"), it is simply ignored and treated the same as "R:".

All the function formats are given in their Atari BASIC form. If using assembly language or some other high level language, refer to the language manual for its equivalent form.

Opening the RS-232 Port

Syntax
OPEN #IOCB,Aux1,0,"R:"

Remarks
This function opens a channel to the RS-232 port in pseudo "non-concurrent" mode. To remain compatible with the 850 and P:R: Connection, the MIO has a flag indicating whether an XIO 40 (set concurrent mode) has been performed since the last OPEN command. If it has not, then the STATUS command returns the state of the handshake lines, whereas if in concurrent mode, the STATUS command returns the number of characters in the input and output buffers.

Aux1 contains the I/O direction bits — 4 for input only, 8 for output only, and 12 for both input and output (which is equivalent to 13 of the 850 interface). Many XIO calls do not require that you open an RS-232 channel first, however, it is generally a good practice to open the channel first. Care should be taken when setting the state of the handshake lines; if you set DTR false, the transmitter and receiver are disabled. Therefore, you must re-enable them by setting DTR true before continuing with normal operation.

When the channel is opened, both the input and output buffers are cleared. Also, the RTS and DTR handshake lines are set true (to the ready state).

Closing the RS-232 Port

Syntax
CLOSE #IOCB

Remarks
This statement closes the IOCB connected to the RS-232 port. This simply shuts down the IOCB; the RS-232 port remains untouched except that the system waits until all data in the output buffer has been transmitted.

Input Character or Line From the RS-232 Port

Syntax
GET #IOCB,varb
INPUT #IOCB,varb$

Remarks
These functions input data from the RS-232 port; the GET statement inputs the numeric value of one character into a numeric variable and the INPUT statement inputs a string of characters into a string variable. On the INPUT statement, if the data is a numerical ASCII string, you may read the data into a numeric variable. Input strings are terminated by an end-of-line (EOL) character.

Note that the IOCB must be opened for read or read/write (modes 4 or 12), but whether in concurrent mode or not has no effect on GET/INPUT statement operation. Refer to your BASIC reference manual for more information on these statements.

Output Character or Line To the RS-232 Port

Syntax
PUT #IOCB,exp
PRINT #IOCB;exp$

Remarks
These functions output data to the RS-232 port; the PUT statement outputs the numeric value of one character to the port, and the PRINT statement outputs a string of characters to the port. The syntax of the PRINT statement is the same as a normal PRINT statement except that the "#IOCB;" precedes the expression.

Note that the IOCB must be opened for write or read/write (mode 8 or 12), but whether in concurrent mode or not has no effect on GET/INPUT statement operation. Refer to your BASIC reference manual for more information on these statements.

Reading the Port Status

Syntax
STATUS #IOCB,DUMMY
FLAGS = PEEK(746) : REM Error bits relating to status history
LINESTAT = PEEK(747) : REM Status of handshake lines
or
STATUS #IOCB,DUMMY
FLAGS = PEEK(746) : REM Error bits relating to status history
INCHARS = PEEK(747) : REM Number of chars in input buffer
OUTCHARS = PEEK(749) : REM Number of chars in output buffer

Remarks
These statement sequences are useful for determining many facts about the state of the RS-232 port. The first syntax is used when in pseudo "block mode I/O" (same as "non-concurrent"), whereas the second is used in concurrent mode I/O. Notice that the variable DUMMY is simply a CIO status of the success of the STATUS command. If there were an error (DUMMY<>1), then BASIC would halt and give an error message (unless a TRAP was performed prior to the STATUS).

The block mode STATUS (first syntax) returns a status history of the port (in FLAGS) and the state of the control lines (in LINESTAT). The meaning of each bit is given in tables 6-1 and 6-2.

The concurrent mode STATUS (second syntax) returns a status history of the port (in FLAGS) and the number of characters in the input buffer (in INCHARS) and in the output buffer (in OUTCHARS). The meaning of each bit of FLAGS is given in table 6-1.

Table 6-1. Meaning of Error Bits From Location 746

Bit	Dec Equiv	Meaning of Error
7	128	Received a data framing error
6	614	Received a data byte overrun error
5	32	Received a data parity error
14	16	Received a buffer overflow error (>255 chars)

Table 6-2. Meaning of Status Bits From Location 747

Bit*	Dec Equiv	Meaning When Bit is Set (1)
7	128	DSR is true (ready)
5	32	CTS is true (ready - Always true on MIo)

3	8	CRX is true (ready)
0	1	RCV is at MARK (Always Set on MIO)

* Bits 6, 4, and 2 are simply copies of the next highest bit. In the 850 Interface, these bits would indicate a history (i.e. not always ready since last STATUS).

Forcing Early Transmission of Output Blocks

Syntax
XIO 32,#IOCB,0,0,"R:"

Remarks
This function causes all the buffered data in the computer to be transmitted through the RS-232 port. This is used when the user wants to make sure that all data is transmitted before performing his next function. (This could also be performed by doing status request until the output data length is zero.)

Controlling Outgoing Lines DTR, RTS, and XMT

Syntax
XIO 34,#IOCB,Aux1,0,"R:"

Remarks
This function allows you to set the state of the output handshaking lines. This function may be perform in both concurrent and "non-current" mode (there is really no difference except for the way STATUS commands are interpreted). Care should be taken when disabling DTR (setting to false) since transmission and receiving are halted until DTR is set TRUE. Aux1 is coded as indicated by table 4-3.

Table 6-3. Control Values Added to Aux1 (XIO 34)

Function	Bit	Dec Equiv	Meaning When Bit is SET
DTR	7	128	Set state of DTR (from bit 6)
	6	64	Set DTR Ready (Not ready if bit is CLEAR)
RTS	5	32	Set state of RTS (from bit 4)
	14	16	Set RTS Ready (not ready if bit is CLEAR)
XMT	1	2	Set state of XMT (from bit 0)
	0	1	Set XMT to MARK (SPACE if bit is CLEAR)

Setting Baud Rate, Stop Bits, and Ready Checking

Syntax
XIO 36,#IOCB,Aux1,0,"R:"

Remarks
This function configures the RS-232 port for desired speed and stop bits. Aux1 is the sum of two codes; baud rate and the number of stop bits. The coding is given by Table 6-4. You must add the value representing the desired baud rate to the code (0 or 128) for the desired number of stop bits per word. Note that the word size is always 8 bits plus 1 or 2 stop bits; the MIO "R:" handler doesnot support smaller word sizes as did the Atari 850 interface.

The "missing" baud rates are available through the ACIA on the MIO but are not supported by the "R:" handler since they are never used.

Table 6-4. Codes to Add to Aux1 (XIO 36)*

Add	Baud Rate	Add	Baud Rate
0	300	5	110
8	300	9	600
10	1200	12	2400
13	9600	14	9600

15	19200

* Default is 1 stop bit. Add 128 for 2 stop bits.

Setting Translation Modes and Parity

Syntax
XIO 38,#IOCB,Aux1,0,"R:"

Remarks
This function configures the parity and level of ASCII/ATASCII translation. The value of Aux1 is derived from Table 6-5.

Table 6-5. Control Values Added to Aux1 (XIO 38)

Function	Add	Besulting Function Performed
PARITY	0	No parity (8-bit data is untouched) (default)
	4	Check/Set odd parity, clear parity bit
	8	Check/Set even parity, clear parity bit
	12	Send mark parity, clear parity bit
TRANS-	0	Light ATASCII/ASCII translation (default)
LATION	32	No translation
LINE	0	Do not append LF after CR (default)
FEEDS	64	Append LF after CR (translated from EOL)

Setting Concurrent Mode

Syntax
XIO 40,#IOCB,0,0,"R:"

Remarks
This function simulates the "Start concurrent mode I/O" of the P:R: Connection and Atari 850. This is needed because of the dual nature of the STATUS command. When in concurrent mode, the statement returns the size of the data buffers, whereas, when in "block mode", the statement returns the state of the handshake lines. This has no effect on the rest of the "R:" handler functioning or on the ACIA. It simply set a flag for the STATUS function.

Chapter VIII

Appendix A - Other ICD Product Offerings

8 Appendix A - Other ICD Product Offerings

R-TIME 8 Clock Cartridge – $69.95 – adding a new dimension to the Atari: TIME/DATE — developed to provide continuous and automatic time/date information — plugs into any cartridge slot on an Atari Computer and updates SpartaDOS automatically and accurately. The R-Time 8 works in the right or left slot of the 800 computer and can be left in either cartridge port on the 130XE adapter board for the MIO. A unique cartridge extension port in the top allows the use of other cartridges when using an XL/XE series computer. The R-Time 8 is supported through software as the clock for Bulletin Board Construction Set (BBCS) from the ANTIC Arcade and now BBS Express from Orion Micro Systems (written by Keith Ledbetter). Our new 'ZHAND' clock handler works with any DOS and allows simple XIO calls for easy R-Time 8 access through your favorite programming language. It comes with a built in 3-5 year battery! The R-TIME 8 does not use any cartridge area memory and is decoded in the $D5B8 - $D5BF range so it will not interfere with other cartridges including Basic XE. ICD will provide complete service for $20 US flat rate which includes all parts, timing adjustment, and battery replacement.

SpartaDOS Construction Set – $39.95 – a special SpartaDOS utility package which includes seven different SpartaDOS versions and many new disk utilities including: RAMDISK support for the 130XE and RAMBO XL, ATR8000 support, RPM test, a high speed sector copier, hard disk drive support, and much, much more. A special menu file allows rapid transfer, erasure, lock or unlock of tagged files, using only the SPACE bar, OPTION, START, cursor and SELECT keys. You want more you say? Ho about a 32 character keyboard buffer, intelligent switching between disk densities, a binary file games menu, subdirectories, time/date file stamping? SpartaDOS is the only DOS with all these features. The new SDCS manual is 175 typeset pages of everything you wanted to know about SpartaDOS and the US Doubler. Also included is the SpartaDOS version 3.2 software and manual for BASIC XE, 1200XL, and hard disk drive support. There is so much software included in this package that we distribute SDCS on two unprotected disks using three sides! Included free are seven arcade quality games.

SpartaDOS, the essence of SDCS, is a totally new advanced DOS for the Atari Computer. SpartaDOS enhances the entire 8-bit Atari Computer line; gives it the power of an Apple or IBM machine. SpartaDOS puts a time/date stamp on each file when it is created or rewritten. This information is then displayed when the directory is listed. SpartaDOS also supports the 1050 "dual density" which means you can use it with your unmodified 1050 in both single and dual density. Our US Doubler allows your 1050 access to all three densities. SpartaDOS also works with the Indus GT, Rana 1000, and other third party drives for the Atari.

SpartaDOS is also memory resident. This means you can go to BASIC then back to DOS without reloading DOS from disk. Special XL/XE SpartaDOS version even gives you an extra 4000 bytes of free memory! Input and output redirection is supported through 'batch files' and the 'print' command. SpartaDOS supports the ATR8000 with 5 1/4 and 8 inch, single or double sided drives, 36 - 80 tracks. An ATR8000 serial communications handler is also provided.

You get SpartaDOS free as part of the SpartaDOS Construction Set or with the US Doubler.

<u>So you want a second or third opinion, then let's talk about reviews.</u>

In his February 1986 ANALOG review of SpartaDOS, Matthew Ratcliff raved, "With this DOS, I can read from and write to any DOS in any density — without any special utilities. This single features makes SpartaDOS the most powerful disk operating system I've ever seen for the Atari XL/XE computer." In October 1985 ANALOG Computing, Art Leyenberger states in his column THE END USER: "There's an old saying that someone will always build a better mousetrap. ICD will certainly be catching more than mice with their new SpartaDOS Construction Set (SDCS)... As I said before: the more I use it, the more I like ICD's SpartaDOS Construction Set. It could easily be the ultimate DOS for the 8-bit Atari computers." In May 1985 ANALOG Computing, Russell Haupert, in his review of the US Doubler from ICD, Inc., raved: "The DOS is very rich in features and a great pleasure to use." Peter Ellison in ROM Magazine (DEC/JAN 85): "SpartaDOS is the best I have ever used... a product so revolutionary it adds new life to the Atari..." What more can we say?

SpartaDOS X – $79.95 This is an all new advanced version of SpartaDOS in our unique piggyback 'X' cartridge. SpartaDOS X is a totally relocatable DOS which allows operation on any 8-bit Atari

with expanded RAM by relocating the DOS overhead in the additional RAM area. This feature of SpartaDOS X gives more free memory to BASIC and other languages operating within the normal 64K boundries of the 6502 CPU. SpartaDOS X supports UltraSpeed I/O on Indus GT drives. SpartaDOS X has the most powerful database ever developed for the Atari. It rivals dBASE III in search speed!

US Doubler with SpartaDOS! – $69.95 True Double Density (180 kilobytes) and high speed I/O for the 1050 drive. Also supports single density and the 130 KB Atari 'Dual Density' or 1050 mode. This is fully compatible with all existing Atari software. The high speed I/O (UltraSpeed) works with SpartaDOS to TRIPLE the speed of data transfer! Now it includes SpartaDOS Construction Set. Two plug-in chips in the 1050 must be replaced and now, no soldering is required in most cases. (Call your dealer for details.) This hardware modification also fixes some of the bugs in the early 1050s that prevented them from loading certain software programs. Even when not using SpartaDOS, the US Doubler increases the 1050s speed by about 8% and reads and writes more accurately than most standard 1050s. The US Doubler makes your 1050 drive compatible with all other DOS's but we are sure you will use SpartaDOS as your main DOS once you learn the natural and powerful command structure. The new US Doubler package includes the SpartaDOS Construction Set manual with an easy to follow installation section, two SpartaDOS program diskettes, and the plug-in hardware modules.

What do the reviews say, you sagely inquire?

In the MAY 1985 issue of ANALOG Computing, Russell Haupert waxes: "In tests, the UltraSpeed I/O worked as stated... The old bleep bleep of POKEY is replaced by a staccato rush that sounds more like machine gun fire... I've tested the US Doubler in all three formats and am happy to report that it performs as promised... Most importantly, using the true double density afforded by this enhancement I've attained compatibility with my friends disks, as well as reducing my disk count by one half." Peter Ellison of R.O.M. Magazine was equally impressed: "If you have a 1050, this is a fantastic deal. It is fully compatible with all software, and allows a whole lot more disk space." In the December 1985 annual shoppers guide, ANTIC, the Atari Resource, selected US Doubler as one of the best products of 1985 in the Unique Hardware category.

US Doubler 1–4 or US Doubler 5–8 – $39.95 – The US Doubler 1-4 is our famous 1050 disk drive modification without the SpartaDOS Construction Set. This includes a complete installation manual and warranty but does not include the technical notes found in the SpartaDOS Construction Set manual. The US Doubler 1-4 is primarily for people who already own the SpartaDOS Construction Set. US Doubler 1-4 will give your Atari 1050 full double density operation with any program designed to support double density on Atari computers.

The 5-8 version was developed for registered US Doubler owners who desire to use more than four drives with their Atari computer. The most obvious use is for bulletin boards. This special US Doubler chip set allows your drive to be set to any drive number from 5 through 8. SpartaDOS is not included since this is for registered US Doubler or SDCS owners only. It works with any DOS which supports more than 4 drives. For operation under SpartaDOS use version 2.3d or later. Note: you will not be able to use this drive as #1 (only 5-8) to boot the system with this special chip set installed.

RAMBO XL – $39.95 – makes your 800XL or 1200XL memory compatible with the 130XE! This internal memory upgrade kit supplies all the signals necessary to turn your XL into a powerful 256K computer. Now you can use powerful BASIC XE with your XL computer! RAMBO XL supports the standard 64K RAMDISK supplied with Atari DOS 2.5. The new RD.COM handler with SpartaDOS Construction Set gives a 192K RAMDISK! That's enough memory to duplicate a full double density disk in one pass! The new PaperClip word processor from Batteries Included supports RAMBO XL. The new Synfile+ database from Synapse/Broderbund now supports this memory expansion! RAMBO XL includes a plug-in decoding board and complete installation instructions. You supply the 8-256K DRAMs which plug into existing sockets where the 64K chips now reside (available for $28.00 direct, price may change). Note: Some 800XLs don't have sockets on all ICs. his will require extensive de-soldering and should only be attempted by a professional. ICD will install and test RAMBO XL on your computer for $30.00 including shipping.

P:R: Connection – $89.95 – a reliable, economical serial/parallel interface for all 8-bit Atari

computers. Includes a built in R: handler. Atari 850 compatible at a much better price.

Printer Cable – $14.95 – between your printer and MIO or P:R: Connection. Cables are necessary to connect your printer to either the MIO or the P:R: Connection. High quality 6 foot cables are available for printers with a 36 pin 'centronics' type connector.

MODEM Cable – $14.95 – between your MODEM and MIO or P:R: Connection. Cables are necessary to connect your MODEM to either the MIO or the P:R: Connection. High quality 6 foot cables are available for MODEMs with the standard DB25S. Other configurations available upon request.

Printer Connection – $59.95 – an economical, high quality printer only interface cable for 'centronics' type parallel printers. This unit is built into a 10 foot cable and has metal locking ears to prevent 'fallout' which is a problem with other brands.

Be sure to register your MIO by filling out and sending in your warranty/update card at the back of this manual. This will make you eligible for a free subscription to our quarterly newsletter, OPEN FILE! This will help to keep you abreast of our latest products and newest designs.

Chapter IX

Appendix B - Standard Printer & MODEM Cables

9 Appendix B - Standard Printer & MODEM Cables

The following two tables are the standard connection specifications used by ICD for our standard printer and MODEM cables. These should work for the most common printers and MODEMs or they may need to be modified according to the special needs of your particular installation.

Printer Cable Connections

36 pin centronics \'male)	DB15P
1	1 - Data Strobe
2	2 - DO
3	3 - Dl
4	4 - D2
5	5 - D3
6	6 - D4
7	7 - D5
8	8 - D6
11	11 - Gnd
12	12 - Fault
13	13 - Busy
15	15 - D7
Frame - to the shield wire	No connect~on to shield

****

MODEM Cable Connections

DB25P	DB9F
2	1.-.DTR
a	2 - CRX
2	3 -.XMT
3	14 - RQV
7	5 - GND
6	6 - DSR
4	7 - RTS
5	8 - CTS
Frame -to the shield wire	No connection to shield

The following table is used with a Toshiba P321 printer with serial interface and is only shown as a guide. All handshaking lines are connected even though most applications would work with only XMT, RCV, and GND. The Toshiba printer is a DTE device and the MIO is also a DTE device. This requires the crossing of XMT and RCV. Be sure to check your printer manual for pin out as most printer connectors will vary. If your serial printer is a DCE device (not DTE) then XMT would connect to XMT and RCV to RCV.

DTE Serial Printer Cable Connections

DB5P	DB9?
6 - DSB	1 - DTR
24 - RTS	2 - CRX
3 - RCV	3 - XMT
2 - XMT	24 - RCV
7 - GND	5 - GND
20 - DTR	6 - DSR
5,8- CTS,CD	7 - RTS
14 - FAULT	8 - CTS
Frame -to the	No connection to shield

```
shield wire
```

Null MODEM Cable Connections

This will allow transfer of files between an MIO serial port or P:R: Connection and an IBM PC or Atari ST without using MODEMs. Tie pins 6 and 8 together at the DB25 end (PC or ST).

```
DB25S                  DB9?
8  - CRX               1 - DTR
20 - DTR               2 - CRX
3  - RCV               3 - XMT
2  - XMT               4 - RCV
7  - GND               5 - GND
     NC                6 - DSR
     NC                7 - RTS
     NC                8 - CTS
Frame - to the         No connection to shield
shield wire
```

Chapter X

Appendix C - Compatibility

10 Appendix C - Compatibility

The following is a list of MODEM software which has been found to be incompatible with the MIO R: handler and our fix or solution to the problem.

TSCOPE by Joe Miller
TSCOPE was designed several years ago as a terminal program to support COMPUSERVE Vidtex (a trademark of COMPUSERVE) file transfer protocol. It only works with the 850 (not the Atari direct connect MODEMs) and is used by many COMPUSERVE subscribers. Joe Miller has graciously modified TSCOPE into RSCOPE which will now support the P:R: Connection and the MIO, as well as the 850. RSCOPE is in the public domain and available for downloading from various boards including our ICD BBS (815/968-2229).

Chapter XI

Appendix D - Hard Drive Data

11 Appendix D - Hard Drive Data

The following table lists some of the more common hard drives and the number of cylinders and heads on the drive. Adaptec controllers usually give 2% more sectors than the calculated max. RLL controllers (like Adaptec 4070) are required for RLL capacities listed, MRLL is an embedded modified RLL controller.

Vendor	Model #	Head	Cyln	Meg	Size	Max # Sectors
Disctron						
	D526	8	306	20	5.25F	78,336
Lapine						
	LT2OOO	4	615	20	3.5H	78,720
	L~C0	4	615	20	3.5H	78,720
	LT3OO	4	615	30	3.5H	118 ,o80
(RLL)						
	LT4OO	4	615	40	3.5H	157,440
(MRLL)						
MicroScience						
	HH6l2B	2	612	10	5.25H	39 ,i68
	HH725A	4	612	20	5.25H	78,336
Microcomputer Memories						
	Ml12	4	306	10	3.5H	39,168
	~12	4	306	10	5.25H	39,168
	M312	4	306	10	5.2SF	39,168
Miniscribe						
	MS3012	2	612	10	5.25H	39,168
	MS3425	4	615	20	5.25H	78,720
	M53438	4	615	30	5.25H	118,080
(RLL)						
	MS8425	4	615	20	3.5H	78,720
	MS8438	4	615	30	3.5H	118 ,080
(RLL)						
	MS6032	3	1024	25	5.25F	98,304
	MS6053	5	1024	42	5.25F	163,840
	M56085	8	1024	67	5.2SF	262,144
	1D40	5	981	40	5.2SF	156 ,960
	1D60	7	981	60	5.2SF	219 ,744
Rodime						
	R0202E	4	640	20		81,920
	B0203E	6	640	32		122 ,88o
	B0204E	8	480	32		122,880
	R0252	4	306	10	3.5H	39,168
C						
	R0652	4	612	20	3.5H	78,336
Seagate						
	ST506	4	153	5	5.25F	19,584
t						
	ST412	4	306	10	5.2SF	39,168
	ST212	4	306	10	5.25H	39,168
	ST213	2	612	10	5.25H	39,168
	ST225	4	615	20	5.25H	78,720
	ST225N	4	615	20	5.25H	78,720
	ST238	4	615	30	5.25H	118,080
(RLL)						
	ST4C26	4	612	20	5.25F	78,336
	ST4030	5	733	32	5.2SF	117,280
	ST4038	5	733	32	5.25F	117 ,280
	ST4C51			40	5.25F	
Shugart						

	5A706	2	306	5	19,584	
	5A712	4	306	10	39,168	
Tandon						
	TM5Ol	2	306	5	19,584	
	TM502	4	306	10	39,168	
	TM252	4	306	10	S . 25H	39,168
Tulin						
	TL226	4	640	20	5.25H	81,920
	TL240	6	640	32	5.25H	122,880
	TL340			32	5.25H	

Chapter XII

Appendix E - Possible Errors Using the MIO

12 Appendix E - Possible Errors Using the MIO

Code #		ERROR CODE MEANING
128	($80)	Break Key Was Pressed
129	($81)	IOCB Already Open
130	($82)	Nonexistent Device
131	($83)	Open for Write Only
132	($84)	Invalid XIO Call Made
133	($85)	IOCB Not Open (from CIO)
135	($87)	Open for Read Only
138	($8A)	Device Timeout
139	($8B)	NAK — Input Handshake Lines Not Ready
153	($99)	Already in Concurrent Mode

Index

- 1 -

- 8 -

- A -

- B -

- C -

- D -

- E -

- F -

- G -

- H -

- I -

- L -

- M -

- N -

- O -

- P -

- R -

- S -

- W -

- X -